FMCGIC FileMaker CGI in C

by Joerg Richter version 1.0.1 2/4/96

Contents

1. Introduction
2. Database requests
3. Installation
4. Preparation
  4.1 HTML form
  4.2 Database fields
5. Running
6. Testing

Appendix

A. Administrator Access
B. Logging
C. Customization
  C.1 HTML generation
  C.2 Error messages
  C.3 Logfile
D. Version History

Used Terms

Administrator
The Administrator is any party running a Macintosh computer as server in the Internet, installing and providing a FileMaker database along with FMCGIC. This Manual is adressed to administrators.

User
User is any party using a computer linked to the Internet, accessing the database provided by the administrator on any remote server through a WWW Browser.

1. Introduction

FMCGIC is a CGI application (complying with the Common Gateway Interface standard) for Macintosh computers allowing communication between a HTTP Server and FileMaker database program. Using FMCGIC the HTTP Server administrator will be able to display HTML pages in the Internet that are created dynamically by the user's request to the database. After FMCGIC is installed, Internet Users will be able to retrieve information from any database offered on the server through a WWW Browser.

FMCGIC features:

- **Fast** — only 1 second average duration for complete database requests
- **Universal** — FMCGIC works with any FileMaker database, also with multiple databases simultanously
- **Adaptable** — multiple page layouts for the pages created by FMCGIC can be defined
- **User friendly** — the user can limit the number of records shown per page. If more records are found, a "Next xx records" link occurs automatically.
- **Compact** — FMCGIC requires 250K memory and 48K disk space (PPC: 384K resp. 64K)

FMCGIC was tested with the MacHTTP 2.2 server software. It may work with other servers, but this has not been tested to date. FMCGIC comes in versions for both 68K and PPC Macintoshes.

FMCGIC was compiled with Metrowerks CodeWarrior 7 using Grants Framework and AEGizmos (great tools). FMCGIC works similarly to Russell Owen's FileMaker CGI, also a well designed CGI — though owing to its AppleScript nature far too slow for acceptable reply times. A complete database request using AppleScript took 10 to 15 seconds on a Centris 660AV but only one second using FMCGIC.

Technically speaking, FMCGIC handles requests originating from a HTTP Server and generates reply pages resulting in HTML text. The current Version of FMCGIC handles three types of server requests, called 'actions'. The first two actions are requests to FileMaker, described in the following chapter 2, "Database Requests". Appendix A, "Administrator Access" deals with the third action.

2. Database Requests

A complete database request and FMCGIC's role works as follows: The user composes a database request on the form displayed in his WWW Browser and uses a button to start the search action. The WWW Browser then transmits the user's input to the HTTP Server specified, which passes it on to FMCGIC. FMCGIC then analyzes the data, queries FileMaker accordingly (through AppleEvents), and waits for the result. Results of Filemaker requests are then composed by FMCGIC as HTML pages and sent back to the HTTP server, who finally sends the resulting page back to the WWW Browser for viewing.

The current FMCGIC version offers two request options, called actions. The **FIND** action lists all records that comply with the user's request. The result of the request is a list of entries, each giving a brief description of the respective record found. Entries are Hyperlinks as well, enabling the user to access the total content of the actual record. The **GET** action handles this access to the complete content of a individual record.

Typically, during a complete database request both actions are involved. Thus, in case too many records are found following a database request, only a manageable amount of data is displayed to the user. If for example a database request results in 30 records found (through the FIND action) the user gets a list of 30 entries for a start. This allows efficient reading, because entries only contain the most important information of the records under consideration. Since all entries are Hyperlinks, the user has direct access to the actual content of the selected data records (through the GET action).

Alternatively to this bipartite request type the contents of all records found could be listed completely. Depending on the database, this could lead to an enormous output of data, difficult to handle by the user. Still, this method is viable if records contain only small amounts of information — and supported by FMCGIC.

3. Installation

FMCGIC is installed best on the same Macintosh that runs MacHTTP and FileMaker. If for any reason FMCGIC should not be installed within the directory which contains MacHTTP, an alias of FMCGIC must be put within the MacHTTP directory.

When implementing Administrator Access (see appendix A) the file "FMCGICAdmin.html" must be deposed within the MacHTTP directory. In addition, a realm should be defined for this page in "MacHTTP.config" providing password protection for this page and entitling only the Administrator to access it. Another realm should be fixed for logfiles created by FMCGIC (see appendix B, "Logging"). FMCGIC uses the format `FMCGIC logfile` *`MM/JJ`* or *`M/JJ`*.

To use the sample database described in the following chapter the files "Default.html", "Hinweise.html" ,"FMCGICAdmin.html" along with FMCGIC program file or an alias must be put in the same directory. Now access to the sample database can be tried (to continue see chapter 5, "Running"). The preparation of FMCGIC for any individual database is described in the following chapter.

4. Preparation

So far FMCGIC has no user interface except the About Box and needs no configuration files. FMCGIC is controlled by two settings: 1. Set-up of the HTML form specifying the database request and 2. Set-up of special fields in the appropriate database.

The sample database named "Videos" demonstrates these two preparatory steps. It can be used for handling video cassettes. The "Videos" database contains the fields: `Director`, `Title`, `Original title`, `Category`, `Remarks`, `Country`, `Year`, `Duration` and `Cassette nr.`

Appendix C, "Customization" describes how to use Resources for creating HTML.

Chapter 2, "Database requests" should be read thoroughly for a better understanding of this comparatively dificult chapter.

4.1 HTML form

The HTML page containing the users' request form has to be composed first. Here is a definition of the request form for the sample database. (The single form components are explained after):

```
<FORM ACTION="FMCGIC.acgi" METHOD="post">

<INPUT TYPE="hidden" NAME="_action"   VALUE="FIND">
<INPUT TYPE="hidden" NAME="_database" VALUE="Videos">
<INPUT TYPE="hidden" NAME="_field"    VALUE="HTMLLink">

      Director:<INPUT TYPE="text" NAME="Director"       size=45>
         Title:<INPUT TYPE="text" NAME="Title"          size=45>
Original title:<INPUT TYPE="text" NAME="Original title" size=45>

Category:<INPUT TYPE="radio" NAME="Category" VALUE="Fiction movie"> Fiction movie
         <INPUT TYPE="radio" NAME="Category" VALUE="Documentary">   Documentary
         <INPUT TYPE="radio" NAME="Category" VALUE="Shortfilm">     Shortfilm
         <INPUT TYPE="radio" NAME="Category" VALUE="Feature">       Feature
         <INPUT TYPE="radio" NAME="Category" VALUE="" CHECKED>      doesn't matter

       Country:<INPUT TYPE="text" NAME="Country" size=14>
          Year:<INPUT TYPE="text" NAME="Year"    size=7>

Restrict answer to <SELECT NAME="_maxRecords">
                       <OPTION> 10
                       <OPTION> 20
                       <OPTION> 30
                       <OPTION> 40
                   </SELECT> movies

<INPUT TYPE="Submit" NAME="Submit" VALUE="Search">
<INPUT TYPE="Reset" VALUE="Reset">

</FORM>
```

<FORM> tag

A form definition begins with the `<FORM>` tag. It has two parameters `ACTION` and `METHOD`. `ACTION` specifies the CGI application processing the input data typed into the form by the user later (here `"FMCGIC.acgi"`). If it is not in the same directory as the HTML page containing the form a path relative to the MacHTTP directory can be specified for the CGI. In the case of FMCGIC the default for `METHOD` is `"post"`.

"Hidden" fields

Three `"hidden"` fields named `"_action"`, `"_database"` and `"_field"` must be defined after the FORM tag (no particular order is required). The `VALUE` parameter informs FMCGIC which action to carry out (here `"FIND"`) , about the database title (here `"Videos"`) and which database field to access (here `"HTMLLink"` — see subsection "A field for the FIND action" of following section 4.2, "Database fields").

Input fields

Input field definitions follow the hidden fields. Text input fields, radio buttons, and popup menues can be used. The defined `NAME` parameter has to match a field name of the FileMaker database. Radio buttons in

HTML forms correspond well with "Selection" fields in FileMaker databases (mark "Use Preselection" option for field definition). The `VALUE` parameter of a radio button definition should correspond to one of the preselection entries.

Limit search results

Additionally a special field named `"_maxRecords"` can be defined. This field defines an upper limit on the number of returned data records. For this the sample uses a popup menu. No option should allow the display of more than 40 data records, because such results are likely to confuse the user. On the other hand FMCGIC created HTML page size is limited to 8K. If no input field named `_maxRecords` is defined, returns are limited to 40 data records.

Buttons

A HTML form needs a submit button to start database access (type `"Submit"` and always named `"Submit"`). The `VALUE` parameter contains the text message displayed in the button (here `"Search"`). A reset button to delete all form input can be added. This reset button has no `NAME` parameter (type `"Reset"`). Again, the `VALUE` parameter contains the text message to be displayed in the buton (here `"Reset"`).

In summary, the four predefined input field (hidden fields also) names are: `_action`, `_database`, `_field` and `_maxRecords`. These names must not to be used as FileMaker database field names. This limitation is minimized by the leading `_` in these predefined names.

4.2 Database fields

In creating HTML FMCGIC uses the assistance of FileMaker. Dependending on the current action it accesses special database fields that already contain fragments of HTML. This is achieved through adding two formula fields (type "Text") for each of the actions to the database by which FileMaker "computes" the corresponding HTML expressions for each data record. Any name can be used for these two fields (exception: see the preceding paragraph). For better control these fields can be included in the database layout, though this is not necessary.

A Field for the FIND action

When the user clicks on the "Start Search" Button in the request form, FMCGIC is called for the first time (see subsection "Buttons" of the preceding section). In this case, the FIND action is invoked returning all data records matching user requests. The resulting page is a list of hyperlinks. Input field contents are passed to FMCGIC as parameters when it is invoked. Input fields are those completed by the user plus "hidden" fields (see subsection "Hidden fields" of the preceding section).

The Find action requires the parameters `_database` and `_field` to tell FMCGIC which database and which field to access. For any data record found during the FIND action, one field is accessed that contains a hyperlink as an HTML fragment. This link leads to the page showing the total content of the data record. Because the page has to be created dynamically, the hyperlink reference again has to contain a FMCGIC call.

In the sample database the field used during the FIND action is called `HTMLLink`. Here is the definition for this field:

field name: "HTMLLink", field type: formula , result type: "Text", non-multiple field, formula:

```
"<LI><A HREF=""FMCGIC.acgi?_action=GET&_database=Videos&" &
"_field=HTMLRecord&Title=" & Title & """>" & Title & "</A>, " & Director & ", " &
Category
```

Using this formula FileMaker might "compute" for the content of the record field HTMLLink of the fiction movie "Good Fellas" by Martin Scorsese the following test expression:

```
<LI><A
HREF="FMCGIC.acgi?_action=GET&_database=Videos&_field=HTMLRecord&Title=Good
Fellas">Good Fellas</A>, Martin Scorsese, Fiction movie
```

This is a regular HTML expression displayed later by the users' WWW Browser as follows:

• Good Fellas, Martin Scorsese, Fiction movie

If the user searches for movies by Scorsese for example, he will get a list of movies that contain entries of this type. Clicking on the hyperlinks thus created, the user starts FMCGIC again to request the total content of the relevant record from the database. Again the hyperlink reference has to contain a call to FMCGIC using the GET action this time. Call-up parameters are now part of the hyperlink reference (and do not result from a HTML form as with the FIND action).

Thus, the hyperlink reference definition (`HREF`) begins with `FMCGIC.acgi`, followed by a `?` marking the beginning of the parameters. The first three parameters `_action`, `_database` and `_field` define the fields necessary for the GET action (see following subsection "A Field for the GET action"). In this example, FMCGIC takes the title of the current data record as search criterium in

the fourth parameter.

In the example, the movie title is the only search criterium. This would not be a unique reference under real world conditions. If, during the GET action, several or no record(s) are found, this points to an error in the hyperlink reference definition. In this case, an error message is displayed to the user.

With regards to FileMaker syntax, it should be remarked that constant text strings must be enclosed in quotation marks. If within the text string a quotation mark is needed, it must be written as `""`. The `&` character is the operator for concatenating text strings and must not be confused with the `&` characters shown in the text strings of the sample displayed above. These & characters originate from CGI standard and are used for separating parameters.

The formula must contain the text strings `<A HREF="", """>"` and `</A>` in any case because they mark the different components of the hyperlink relevant for internal processing in FMCGIC. If one or more of these "marker" character strings are omitted from the formula, an error page is generated by FMCGIC accordingly.

More text can be added to the formula optionally before and after the hyperlink reference. In the sample `<LI>` is used to generate a leading • before the movie title. On the other hand `", "& Director &", " & Category`, list the movie's director and category.

A Field for the GET action

If the user clicks on a hyperlink generated by a FIND action, FMCGIC is called up starting a GET action this time. Parameters for this call are now components of the hyperlink reference of the chosen link. Like the FIND action, the GET action also requires the two parameters `_database` and `_field`. The GET action is supposed to access one record field containig the total content of this record as an HTML fragment.

In the sample database the field used during the GET action is called `HTMLRecord`. Here is the definition for this field

field name: "HTMLRecord", field type: formula, result type :"Text", non-multiple field, formula:

```
"Director: " & Director & "<BR>¶" &
"Title: " & Title & "<BR>¶" &
"Original title: " & Original title & "<BR>¶" &
"Category: " & Category & "<BR>¶" &
"Remarks: " & Remarks & "<BR>¶" &
Country & ", " & Year & "<BR>¶" &
"Duration: " & Left(TimeToText(Duration), 4) & "<BR>¶" &
"Cassette nr.: " & Cassette nr. & "<BR>¶"
```

For the movie "Good Fellas" by Martin Scorsese. FileMaker "computes" the following text expression:

```
Director: Martin Scorsese<BR>
Title: Good Fellas<BR>
Original title: <BR>
Category: Fiction movie<BR>
Remarks: <BR>
USA, 90<BR>
Duration: 2:20<BR>
Cassette nr.: 1<BR>
```

The users' WWW Browser displays this HTML fragment as follows:

Director: Martin Scorsese
Title: Good Fellas
Original title:
Category: Fiction movie
Remarks:
USA, 90
Duration: 2:20
Cassette nr.: 1

Character strings in quotation marks can be chosen arbitrarily and are displayed exactly like this in the HTML page generated. Field names (between `&` characters) indicate fields in the FileMaker database and can be chosen from the list of field names directly. The `¶` character represents a carriage return. It is not relevant for HTML generation but is convenient if the generated HTML source is viewed in a text editor. `Left` and `TimeToText` are FileMaker functions and were used to convert the field "duration" (field type "time") into a text expression. The `Left` function then cuts off the seconds.

5. Running

First the FileMaker database in question has to be opened and the HTTP Server to be started. FMCGIC does not need to be started; the first database request starts it automatically. FMCGIC does not need to be supervised, but stays open and handle requests from HTTP Server as long as it is not terminated either by Administrator Access (see appendix A) or manually direct on the Server.

In its current state, FMCGIC cannot open the database (or FileMaker) automatically. If the database is not open during the first incoming request an error message is sent to the user (generated by FMCGIC in HTML).

FMCGIC checks all internal processes comprehensively for errors. In case of an error, it generates an error message accordingly. A system beep is emitted on the server machine when an error occurs. The detected error is logged in the logfile (see appendix B, "Logging").

6. Testing

A CGI application can only be called by a HTTP server that passes the HTML form parameters to the CGI. Thus, if the HTML page along with the form is not delivered by a server, but loaded from the local harddisk, the CGI will not work correctly nor return anything.

However, to test FMCGIC offline (without involving an HTTP server) three scripts named "FIND test", "GET test" and "QUIT test" are provided. They can be started from ScriptEditor, after the "Videos" sample database was opened. The result of the database request is then displayed in the result window of ScriptEditor as HTML text and can be viewed through a WWW browser. During the test run the database request parameters cannot be supplied by the HTML form but are integrated into these test scripts.

A. Administrator Access

Through the Administrator Access feature, FMCGIC can be terminated from a remote computer for convenient updating on the server machine. This makes sense if the server is located at a remote site (for example, an internet service provider). After FMCGIC is terminated, it can be replaced by the administrator via FTP.

FMCGIC is terminated by calling FMCGIC again by selecting the QUIT action. To trigger a QUIT action, a separate HTML page named "FMCGICAdmin.html" is provided. This page defines a form consisting of the "Quit FMCGIC" button. No further parameters are needed for the QUIT action besides the `_action` parameter (set to `"QUIT"` here).

To ensure that only the administrator has access to this page, a realm can be established (see chapter 3, "Installation").

B. Logging

FMCGIC logs all database requests carried out automatically, along with any errors detected. A separate logfile is created monthly. The logfile for March ’95 for example, is named "FMCGIC logfile 3/95". Logfiles are BBEdit text documents and are stored in the directory containing FMCGIC.

Logfiles are closed after every database request, so they should be readable with a text editor at any time. To exclude user access to logfiles a realm definition should be inserted into "MacHTTP.config".

The logfile logs the HTTP server request completely (user and other input), the number of records found and further statistical data. This is a typical logfile header:

```
This logfile is created by FMCGIC 1.0.1
========================================

FMCGIC 1.0.1 launches at Mon, 1. Apr 1996, 1:00:54 Uhr
======================================================

Server request at Mon, 1. Apr 1996, 1:01:55 Uhr

path_args        = ''
http_search_args = ''
username         = ''
password         = ''
from_user        = ''
client_address   = 'ppphome2.prz.tu-berlin.de.'
post_args        =
'_action=FIND&_database=Videos&_field=HTMLLink&_layout=0&Director=Scorsese&Title=
&Original+title=&Categorie=&Country=&Year=&_sortField=&_maxRecords=10&Submit=Sear
ch'
method           = HTTP_post
server_name      = 'lisa.prz.tu-berlin.de.'
server_port      = 80
script_name      = '/Video/FMCGIC.acgi'
content_type     = 'application/x-www-form-urlencoded'
referer          = 'http://ppphome2.prz.tu-berlin.de/Video/Default.html'
user_agent       = 'Mozilla/2.01 (Macintosh; I; 68K)'
connection       = 0
totalFields      = 13
    1. '_action': 'FIND'
    2. '_database': 'Videos'
    3. '_field': 'HTMLLink'
    4. '_layout': '1'
    5. 'Director': 'Scorsese'
   12. '_maxRecords': '10'
   13. 'Submit': 'Search'
2 record(s) found.
Generated HTML result has 343 characters.
Formular processing: 0.02 secs.
Database access    : 0.68 secs.
HTML generation    : 0.02 secs.
----------------------------------------
```

C. Customization

Any page created by FMCGIC is adaptable in layout to concrete individual demands. Besides the options described below, localized versions of FMCGIC are possible.

Customization options can be divided into three areas:

1. HTML generation

HTML fragments can be added to the page containing the search results. This can be used to add headers and footers. Additionally, pages sent to the user with a negative search result or an error can be modified.

2. Logfile

Formulation of logfile data can be modified. This concerns the Server operator only.

3. Error messages

Modification of error messages returned to the user is possible.

All modifications are performed by editing FMCGIC's STR# and TEXT resources. A resource editor (e.g. ResEdit) is required. Some resources strings contain placeholders. These placeholders are filled with text during generation of the final HTML page by FMCGIC. Placeholders have a leading %. The number, order and types of placeholders in any string must not be changed in any case. Text not representing a placeholder can be altered arbitrarily. The resources are described in detail in the following:

C.1 HTML generation

Customization of HTML generation is done via TEXT resources #200, #300, …, #900, which are described below. FMCGIC's default string is also shown (in red).

#200 — FIND action Result Page Header

FMCGIC uses this string as a basis for generating the HTML page that contains the result of a FIND action (see chapter 2, "Database requests"). Therefore, this string should at least define the HTML page title and page "body" introduction. The HTML fragment created dynamically from the database is appended to this string.

```
<TITLE>Found Records</TITLE>
<BODY>
%ld record(s) found:
<P>
```

This string must contain exactly one `%ld` placeholder, substituted by the number of records found (`%ld` stands for a 32-bit decimal value).

#300 — GET action Result Page Header

FMCGIC uses this string as basis for generating the HTML page containing the results of a GET action (see chapter 2, "Database requests"). Therefore, this string should define at least one HTML page title and introduce the page "body". The HTML fragment created dynamically from the database is appended to this string.

```
<TITLE>Found Record</TITLE>
<BODY>
```

This string must not contain any placeholder (GET action always finds exactly one record).

#400 — Search Result Limitation

FMCGIC puts this string before the HTML fragment created dynamically from the database, if more records are found than displayed (see section 4.1, "HTML form", subparagraph "Search Result Limitation").

```
(Records %d-%d are shown)
<P>
```

This string must contain exactly two `%d` placeholders, substituted by the running numbers of the records shown (`%d` represents a simple decimal value).

In the WWW browser, the default string leads to a display in this fashion:

(Records 1-10 are shown)

#500 — Link to the next records

FMCGIC appends this string to the HTML fragment created dynamically from the database, if more records are found than displayed (see section 4.1, "HTML form", subparagraph "Search Result

Limitation"). This string must contain the template of a hyperlink reference.

```
<P>
<A HREF="%s">Next %d records</A> (%d-%d)
```

This string must contain exactly the four placeholders `%s, %d, %d, %d` (in this order), substituted by the URL the number an the index numbers of the records from the next page.

In the WWW browser, the default string leads to a display in this fashion:

Next 10 records (11-20)

#600 — Result Page Footer

FMCGIC appends this string to the HTML fragment created dynamically from the database. It concludes the result page created, i.e. the page "body".

```
<P>
The database access takes %.2f seconds.
</BODY>
```

This string must contain exactly one `%.2f` placeholder, substituted by the amount of time taken by FMCGIC for the proper database request. The expression `.2f` represents a floating point value (two decimal places). Actually, instead of `2` a different value can be declared, but this does not make sense, since time measurement accuracy amounts to 1/60 sec only.

#700 — Negative Search Result

This string contains the complete HTML description of the page generated if no record matching the user's request is found.

```
<TITLE>No record found</TITLE>
<BODY>
<H2>No records found!</H2>
To widen your search, try to fill in lesser search fields.
</BODY>
```

This string must not contain any placeholder.

#800 — Error Page

This string contains the complete HTML decsription of the page created in case of an error.

```
<TITLE>CGI error</TITLE>
<BODY>
<H2>%s</H2>
%s
</BODY>
```

This string must contain exactly two `%s` placeholders (`%s` represents a string). The first `%s` is substituted with the error type and the second `%s` with the error message (see section C.2, "Error messages").

#900 — Administrator Access Result Page

This string contains the complete HTML description of the page created, in case FMCGIC is terminated by Administrator Access (see appendix A).

```
<TITLE>FMCGIC admininstrator access</TITLE>
<BODY>
<H2> FMCGIC 1.0.1 has been closed now. </H2>
</BODY>
```

This string must not contain any placeholder.

C.2 Error messages

Besides the description of customizing the error messages, this section is also intended to describe the error messages themselves.

The STR# resources #128 and #129 can be used to customize the names of the error classes and the error messages themselves (see appendix C.1, "HTML generation", TEXT resource #800). The STR# resources are described below. FMCGIC's default string is also listed (in red).

Error classes

The STR# resource #128 contains 6 strings, used to name the error classes. This strings must not contain any placeholder.

#1 — `FMCGIC error`

This error occurs, if there is only a demo version of FMCGIC installed on the server machine, and a database request is coming from a remote machine. The demo version is restricted in the way, that it handles database requests only if the WWW client and server are running on the same machine.

#2 — `Form error`

This class comprises errors caused by a wrong definition of the HTML form (see section 4.1, "HTML form").

#3 — `Database error`

This class comprises errors caused by a wrong definition of the database field for the FIND action (see section 4.2, "Database fields").

#4 — `Range error`

This class comprises errors caused by a overflow of FMCGIC's internal buffers (see error class "Range" below).

#5 — `AEGizmos error`

This class comprises errors occured inside the AEGizmos utility, used by FMCGIC. This unlikely situation is not influenced by the user nor the administrator. The author of FMCGIC should be informed of this error.

#6 — `System error`

This class comprises errors occured inside the Macintosh system software. This unlikely situation is not influenced by the user nor the administrator. The author of FMCGIC should be informed of this error.

Error messages

The STR# resource #129 contains 24 strings containing all the error messages FMCGIC can generate. For a description of placeholder types see section C.1, "HTML generation".

Error class "**FMCGIC**"

#1 — `There is only a demo version of FMCGIC installed on the server machine. No remote access allowed. Please inform the administrator of this pages.`

The demo version does allow database access, only if the WWW client and server are running on the same machine.

The error message string must not contain any placeholder.

Error class "**Form**"

#2 — `One or more of the 3 required tokens are missing: _action, _database, _field.`

This error occurs if one or more of the three "hidden" field definitions for the HTML form are missing: _action, _database or _field (see section 4.1, "HTML form").

The error message string must not contain any placeholder.

#3 — `Unknown token: '%s'.`

This error occurs if an input field definition for the HTML form uses a name with a leading underscore but is not one of the following four: _action, _database , _field and _maxRecords (see section 4.1, "HTML form").

The error message string must contain exactly one `%s` placeholder, substituted by the rejected name.

#4 — `Unknown action: '%s'.`

This error occurs if the HTML Form input field definition `_action` contains an unexpected action as VALUE component. Only the following three actions are supported: FIND, GET and QUIT (see section 4.1, "HTML form").

The error message string must contain exactly one `%s` placeholder, substituted by the unsupported action.

#5 — `Illegal layout: '%d'.`

This error occurs if the HTML Form input field definition `_layout` contains an invalid layout number as VALUE component. For information about the use of multiple layouts, see the associated file "FMCGIC 1.0.1 README!"

The error message string must contain exactly one `%d` placeholder, substituted by the invalid layout number.

Error class "**Database**"

#6 — `'<A HREF="" ' is missing in link reference: '%s'.`

This error occurs if the link reference definition of the database field for a FIND action contains a syntax error (see section 4.2, "Database fields").

The error message string must contain exactly one `%s` placeholder, substituted by the incorrect field contents.

#7 — `Trailing '"">' is missing in link reference: '%s'.`

This error occurs if the link reference definition of the database field for a FIND action contains a syntax error (see section 4.2, "Database fields").

The error message string must contain exactly one `%s` placeholder, substituted by the incorrect field contents.

#8 — `'</A>' is missing in link reference: '%s'.`

This error occurs if the link reference definition of the database field for a FIND action contains a syntax error (see section 4.2, "Database fields").

The error message string must contain exactly one `%s` placeholder, substituted by the incorrect field contents.

#9 — `Got more than one record while GET action.`

This error occurs if the database field for the FIND action contains a FMCGIC call which do not refer to a unique record (see section 4.2, "Database fields").

The error message string must not contain any placeholder.

Error class "**Range**"

#10 — `AEGizmos description is going too large while adding a comparison descriptor record: %ld chars, maximal %ld chars allowed.`

This error occurs if the description of the AppleEvent to be created by FMCGIC for requesting the FileMaker database is getting too complex. This could be the case if the user fills out many request form fields.

The error message string must contain exactly two `%ld` placeholders, substituted by the size of the description originating and the maximum size allowed.

#11 — `AEGizmos description is going too large while adding the tail: %ld chars, maximal %ld chars allowed.`

This error occurs if the description of the AppleEvent to be created by FMCGIC for requesting the

FileMaker database is too complex. This could be the case if the user fills out many request form fields.

The error message string must contain exactly two `%ld` placeholders, substituted by the size of the description originating and the maximum size allowed.

#12 — `The content of the field '%s' is too large: %ld chars, maximal %ld chars allowed.`

This error occurs if the content of a requested database field is too large. There is a limit of 2K imposed.

The error message string must contain exactly the Placeholders `%s`, `%ld`, `%ld` (in this order), substituted by the name of the requested field, the size of the content and the maximum size allowed.

#13 — `Text between two subsequent link references of the content of field '%s' is too large: %ld chars, maximal %ld chars allowed.`

This error occurs if the text between two subsequent link references of the contents of the field for the FIND action is too large. There is a limit of 2K imposed.

The error message string must contain exactly the Placeholders `%s`, `%ld`, `%ld` (in this order), substituted by the name of the requested field, the size of the text and the maximum size allowed.

#14 — `Generated HTML is going too large while adding a link: %ld chars, maximal %ld chars allowed.`

This error occurs if the HTML page to be created by FMCGIC as the result of a FIND action is too large. This can be the case if too many records are allowed for display (see section 4.1, "HTML form", subparagraph "Limiting Search Result") or if the content of the FIND action database field is very extensive (see section 4.2, "Database fields").

The error message string must contain exactly two `%ld` placeholders, substituted by the size of the HTML page originating and the maximum size allowed.

#15 — `Generated HTML is going too large while adding a record: %ld chars, maximal %ld chars allowed.`

This error occurs if the HTML page to be created by FMCGIC as the result of a GET action is too large. This could be the case if the record found is very extensive.

The error message string must contain exactly two `%ld` placeholders, substituted by the size of the HTML page originating and the maximum size allowed.

#16 — `Generated HTML is going too large while adding the link to next records: %ld chars, maximal %ld chars allowed.`

This error occurs if the HTML page to be created by FMCGIC as the result of a FIND action is too large. This can be the case if too many records are allowed for display (see section 4.1, "HTML form", subparagraph "Limiting Search Result") or if the content of the FIND action database field is very extensive (see section 4.2, "Database fields").

The error message string must contain exactly two `%ld` placeholders, substituted by the size of the HTML page originating and the maximum size allowed.

#17 — `Generated HTML is going too large while adding the footer: %ld chars, maximal %ld chars allowed.`

This error occurs if the HTML page to be created by FMCGIC as result of a database request is too large. See the preceding error messages for possible reasons (for FIND and GET actions, respectively).

The error message string must contain exactly two `%ld` placeholders, substituted by the size of the HTML page originating and the maximum size allowed.

Error class "**AEGizmos**"

#18 — `Error in AEBuildAppleEvent: %d.`

This error message results when an error occurrs within the "AEGizmos" utility used by FMCGIC. This unlikely situation is not influenced by the user nor the administrator. The author of FMCGIC should be informed of this error.

The error message string must contain exactly one `%d` placeholder, substituted by an error code.

Error Class "**System**"

#19 — `Error while sending an Apple event: probably FileMaker dosn't run on the server machine.`

This error occurs if transmission of the AppleEvent from FMCGIC to FileMaker was unsuccessful. Presumably FileMaker was not opened on the server. The Administrator should be informed of this error.

The error message must not contain any placeholder.

#20 — `Error in AEGetParamDesc: %d.`

This error message results from an error occurred within Macintosh system software. This unlikely situation is not influenced by the user nor the administrator. The author of FMCGIC should be informed of this error.

The error message string must contain exactly one `%d` placeholder, substituted by an error code.

#21 — `Error in AECountItems: %d.`

This error message results from an error occurred within Macintosh system software. This unlikely situation is not influenced by the user nor the administrator. The author of FMCGIC should be informed of this error.

The error message string must contain exactly one `%d` placeholder, substituted by an error code.

#22 — `Error in AEGetNthPtr: %d.`

This error message results from an error occurred within Macintosh system software. This unlikely situation is not influenced by the user nor the administrator. The author of FMCGIC should be informed of this error.

The error message string must contain exactly one `%d` placeholder, substituted by an error code.

#23 — `Error in FSWrite: %d.`
This error message results from an error occurred within Macintosh system software. This unlikely situation is not influenced by the user nor the administrator. The author of FMCGIC should be informed of this error.

The error message string must contain exactly one `%d` placeholder, substituted by an error code.

#24 — `Error in FSClose: %d.`

This error message results from an error occurred within Macintosh system software. This unlikely situation is not influenced by the user nor the administrator. The author of FMCGIC should be informed of this error.

The error message string must contain exactly one `%d` placeholder, substituted by an error code.

C.3 Logfile

Formulation of certain data protocolled by FMCGIC in the logfile can be modified via STR# resource 130 (see appendix B, "Logging"). It contains 5 strings described below. FMCGIC's default string is also listed (in red). For a description of placeholder types see the preceding section.

#1 - Logfile Header

This string introduces every logfile created by FMCGIC.

```
This logfile is created by FMCGIC 1.0.1
=======================================
```

This string must not contain any placeholder.

#2 Database Request Time

This string logs database request times. All data transmitted from the HTTP server to FMCGIC are appended to this string (these data include, among other things, all input made by the user in the form.

```
Server request at %s
```

This string must contain exactly one `%s` placeholder, substituted by date and time of the database request.

#3 - Statistical Data about a Database Request

This string is protocolled in the logfile according to server request data. It is used for formatting certain statistical data regarding the logged database request.

```
%ld record(s) found.
Generated HTML result has %ld characters.
Formular processing: %.2f secs.
Database access    : %.2f secs.
HTML generation    : %.2f secs.
----------------------------------------
```

This string must contain exactly the five placeholders `%ld, %ld, %.2f, %.2f, %.2f` (in this order), substituted by the following data: 1. number of records found, 2. size of the HTML page created, 3. amount of time, in seconds, taken by FMCGIC for processing the data transmitted from the HTTP server (these include user input on the form), 4. amount of time, in seconds, taken by FMCGIC for the proper database request and 5. amount of time, in seconds, taken by FMCGIC for creating the resulting HTML page.

#4 - FMCGIC Start Time

This string is protocolled in the logfile when FMCGIC is started (see chapter 5, "Running").

```
FMCGIC 1.0.1 launches at %s
=======================================================
```

This string must contain exactly one `%s` placeholder, substituted by FMCGIC start time and date.

#5 - FMCGIC Closing Time

This string is protocolled in the logfile before FMCGIC terminates itself (see appendix A, "Administrator Access" and chapter 5, "Running").

```
FMCGIC 1.0.1 quits at %s
========================================================
```

This string must contain exactly one `%s` placeholder, substituted by FMCGIC termination time and date.

D. Version History

1.0d1 11/7/95

Primary goal achieved (FIND and GET actions).
Error code handling (with HTML generation).
No consideration of fiels `_sortField` and `_maxRecords` yet.
Built 68K verion with CW6.

1.0d2 11/8/95

Description records disposed.
Consideration of field `_maxRecords`.
Logfile generation.
No more idle time abortion.

1.0d3 11/10/95

Icon layout.
Menu and about box adapted.
Logfiles now contain server request time and more detailed timing data.
Built 68K version with CW7.

1.0b1 11/11/95

Brief documentation (SimpleText).
'Vers' resource splitted for 68K and PPC.
If maxRecords is 0 (no `_maxRecords` field defined) all found records are delivered.
Built PPC version with CW7

1.0b2 11/19/95

Conversion of special characters in all situations.
— Thus fixing database error "get more than one record while GET action".
Default for maxRecords is now 40 (`_maxRecords` field undefined).
Maximum HTML size doubled to 8K.
Minor changes in small icon layout.
Documentation revised.

Sample database changes:

'Duration' field in sample database "Video" is now converted correctly into text.
'Videos' sample database expanded (49 records).
Categories 'documentary' and 'short movie' added.

1.0b3 11/24/95

QUIT action added
FMCGICAdmin.html provided, to trigger the QUIT action.
Test script for the QUIT action provided.
FMCGICs launch and quit times are logged.
'Videos' sample database expanded (119 records).
Documentation revised and expanded by the chapter "Administrator Access"

1.0b4 12/3/95

HTML generation, error and logging messages can be customized by using STR# resources.
A separate logfile is created for every month.
If the search result is limitted for display, this is indicated.

Special character handling now includes typographic quotation marks.

Documentation changes:

Dokumentation revised and considerably expanded.
Chapters "database requests" and "customization" added.
Now in WordPerfect format.

1.0b5 12/8/95

A "Next" link is created automatically, if more records are found than shown. No special configuration is necessary.

1.0b6 1/9/96

For customization of HTML generation now 'TEXT' resources are used instead of 'STR#' resources.
Ballonhelp for the application adapted.

1.0b7 1/11/96

For HTML generation multiple layouts can be used.
The test scripts are made more comfortable.

1.0 1/18/96

Localization (manual and 'TEXT' resources).

1.0.1 2/4/96

In respect of special character conversion in the result of a FIND action, now any number of links are handled (also none).
More extensive checking of internal overflows. Limit for database field contents is 2K, for generated HTML pages 16K.